CHAPTER 4

APPLICATION OF TRADITIONAL MULTI-CRITERIA DECISION MAKING MODELS FOR SELECTION OF PEERS FOR P2P COLLABORATION

Decision making is a process to arrive at a peer or a set of peers for the collaboration activity. The decision making is carried out in two phases. In the first phase, the peer configurations are evaluated and the peers are selected and ranked. The seven steps of the peer collaboration process have been explained in the earlier chapter. It is assumed that the discovery process would obtain the relevant information about the resource configurations of the peers who have agreed to participate in the peer collaboration activities.

The second phase takes place when the peer collaboration actually happens. The set of resources available at a peer does not remain static or always available. These resources may be assigned by the peers for other activities. The resources could also become unavailable due to reasons of failure. If the peers ranked using the static resource configuration are directly selected for running the collaborative application, task execution could fail because of unavailability of resources. It is assumed that the discovery process is re-invoked to provide dynamic update on the available resources at the peers. For each of the peers selected in Phase I, their available resources are once again matched with resources required for the collaborative activity. Peers who have the necessary resources are short-listed and again ranked based on their available resources.
Different MCDM models have been applied for the peer collaboration process and their performance compared Mark et al 2013 and Tomas Gal et al 2013. This chapter discusses the application of traditional MCDM models Analytic Hierarchy Process (AHP) (Saaty T. 1980) (Saaty 1994) and (Saaty 1987), Elimination and Choice Translating Reality (ELECTRE) (Benayoun et al 1966) and Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE) (Brans & Vincke 1985) for the phase I decision making process which is based on peer resource configurations. The feasible set of alternate peers has been determined and the peers have been ranked according to their resource availability. The best or suitable set of alternatives can be chosen for sharing the resources in job execution process.

4.1 ANALYTIC HIERARCHY PROCESS (AHP)

Many multiple attribute decision making methods have been used for ranking in the manufacturing environment. Analytic Hierarchy Process (AHP) remains one of the most popular analytical techniques for selection and ranking (Evangelos and Stuart 1995). Designed to reflect the way people actually think, AHP continues to be the most highly regarded and widely used decision making method. AHP can efficiently deal with objective as well as subjective attributes. AHP has been extensively used in integrated manufacturing (Putrus 1990), in the evaluation of technology investment decisions (Boucher & McStravic 1991), in flexible manufacturing systems (Wabalickis 1988), layout design (Cambron & Evans 1991), risk analysis in transport infrastructure appraisals (Ambrasaitė et al, 2011), energy planning problems (Loken, 2007) (Okeola and Sule 2012) and also in other engineering problems (Wang & Raz 1991). This thesis work has applied AHP for peer selection and ranking for P2P collaboration activity. AHP provides the initiating peer with an overview of criteria, their function at the lower level and goals at the higher level. A further
advantage of AHP is its stability and flexibility regarding changes within and additions to the hierarchy.

AHP is used to rank the participating peers that satisfy the resource criteria. First the resources that are used for ranking are selected based on the needs of the application that has to be executed on the participating peers.

The structure of the decision problem consists of ‘n’ alternatives and ‘m’ decision criteria. Let $a_{ij}$ ($i=1,2,3…\ n$ and $j=1,2,3…m$) represent the performance value of i ’th alternative in terms of j’th criteria. $W_j$ is the weight of the criterian $C_j$. Then pair wise comparison is determined to find the relative importance of each alternative. Pair wise comparisons are done by using a scale which in terms of each criterion is a one to one mapping between the set of discrete linguistic choices available to the decision maker and a discrete set of numbers representing the importance or weight of linguistic choices. This scale is proposed by Saaty 1981. The transformation of original alternate values to a discrete set of numbers is termed as preprocessing.

The next step is to extract the relative importance implied by the previous comparisons. The judgment matrices are formed by pair wise comparisons between the alternatives with respect to each and every criterian. A sample judgment matrix with three criteria is shown in the Table 4.1.

**Table 4.1 Judgment matrix**

<table>
<thead>
<tr>
<th>Attribute 1</th>
<th>Criteria 1</th>
<th>Criteria 2</th>
<th>Criteria 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criteria 1</td>
<td>1</td>
<td>1/5</td>
<td>3</td>
</tr>
<tr>
<td>Criteria 2</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Criteria 3</td>
<td>1/3</td>
<td>1/5</td>
<td>1</td>
</tr>
</tbody>
</table>
Then the corresponding maximum left eigen vector is approximated using geometric mean of every row in the judgment matrix. That is, the elements in each row are multiplied with each other and then the n-th root is taken as n is the number of elements in the row.

If a problem has ‘n’ alternatives and ‘m’ criteria, then the decision maker is required to construct ‘m’ judgment matrices (one for each criterion) of order n x n and one judgment matrix of order m x m (for the ‘m’ criteria).

The numbers are normalized by dividing them with their sum. An evaluation of the eigenvalue approach can be found in (Triantaphyllou and Mann, 1988). An alternative approach for evaluating the relative priorities from a judgment matrix is based on a least squares formulation and is described in (Triantaphyllou et al, 1988).

The consistency index value is needed to be estimated. This is done by adding the columns in the judgment matrix and multiplies the resulting vector by the vector of priorities that is approximated eigenvector obtained earlier. This yields an approximation of the maximum eigenvalue, denoted by $\lambda_{\text{max}}$. Then, the consistency index value is calculated by using the formula: consistency index $= \frac{(\lambda_{\text{max}} - n)}{(n - 1)}$. The consistency ratio is obtained by dividing value by the Random Consistency index.

AHP process for Peer selection is explained using an example. Let the resource attributes considered be CPU speed, (CSp), Number of core Processors (NCore), and Memory size (MSize). Next the relative importance of the attributes has to be decided based on the P2P application that is being collaborated. For instance, the requirement of CPU Speed could be more when the execution time has to be less. Applications that involve more computations may require more number of core processors. All application
would however, typically require multiple heterogeneous resources for efficient execution.

The application of AHP technique for multi-attribute ranking for P2P collaboration is explained and illustrated with an example. Three resources have been considered for this example, namely, CPU speed (CSp), Number of core Processors (NCore), and Memory size (MSize). The AHP decision making process begins with construction of a matrix expressing the relative values of a set of attributes.

The relative importance of these resources is assigned by the decision maker (the initiating peer) based on the requirements of the application. The Saaty rating scale explained in Chapter 3 is used to assign the relative importance to the resources required. AHP is “a theory of measurement through pairwise comparisons and relies on the judgments of experts to derive priority scales” (Saaty, 1982), (Saaty 2008). For the example considered, three ratings have been considered and a number is used to represent the relative importance of the attributes. 1 is used to represent equal importance, 3 for weak importance of one over another and 5 for strong importance. A basic assumption is that if attribute A is absolutely more important than attribute B and is rated at 5, then B must be absolutely less important than A and is valued as 1/5.

The initial matrix for the pair wise comparison is given with the principal diagonal containing entries of 1as each factor is as important as itself. The initial matrix can now be given as in the Table 4.2.
Table 4.2 Initial Matrix is shown as a table

<table>
<thead>
<tr>
<th>Attributes</th>
<th>CSp</th>
<th>NCore</th>
<th>MSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSp</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NCore</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSze</td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Next the Judgment Matrix for each of the resource attributes, Number of core processors, CPU Speed and Memory size is created. The criteria can be fixed based on the importance of the attributes relative to the objective used for peer ranking with respect to cost of resource sharing and ease of operation. The matrix obtained is named as Judgment Matrix. The Judgment matrix for sample attributes like Number of core processors, CPU Speed and Memory size are fixed based on the observed data and is computed as follows.

Considering the number of core processors used for computation, the values could be single, dual or double dual processors. This has been represented as numerical values 1, 2 and 4 in the illustrative example. The dual processor is assumed to be more important than the others hence value 5 is given for dual processors. Single processors are given the next level of importance comparing with dual processors. The double dual processors are assumed to be of weak importance with the value representation of 1. The judgment matrix for number of core processors is shown in Table 4.3.

Table 4.3 Judgment matrix for number of core processors

<table>
<thead>
<tr>
<th>Number of Processors</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1/5</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>1/3</td>
<td>1/5</td>
<td>1</td>
</tr>
</tbody>
</table>
Similarly, the relative importance for the attribute CPU speed is computed by determining the discrete values for a range represented as GHz. Here, the range 2.25-2.5 GHz is considered with the relative importance value to be 5, the range 2.00-2.25 GHz is given a value of 3 and the range 1.75-2.00 has been assigned a value of 1. The judgment matrix for the attribute CPU Speed is given in the table 4.4.

**Table 4.4 Judgment matrix for CPU speed**

<table>
<thead>
<tr>
<th>CPU Speed</th>
<th>1.75-2.00</th>
<th>2.00-2.25</th>
<th>2.25-2.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.75-2.00</td>
<td>1</td>
<td>1/3</td>
<td>1/5</td>
</tr>
<tr>
<td>2.00-2.25</td>
<td>3</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>2.25-2.5</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

The memory size required is categorized as the ranges 1-3 GB, 3-4 GB and 4-32 GB. The relative importance value is computed to be 5 for 3-4 GB, 1 for the range 1-3 GB and 3 for the range 4-32 GB. The judgment matrix for the attribute Memory size is given in Table 4.5.

**Table 4.5 Judgment matrix for Memory Size**

<table>
<thead>
<tr>
<th>Memory Size</th>
<th>1-3</th>
<th>3-4</th>
<th>4-32</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>1</td>
<td>1/5</td>
<td>3</td>
</tr>
<tr>
<td>3-4</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>4-32</td>
<td>1/3</td>
<td>1/5</td>
<td>1</td>
</tr>
</tbody>
</table>

Next, the interdependency among the attributes is determined and is represented using the matrix termed as Overall Preference Matrix shown in Table 4.6.
Table 4.6 Overall Preference Matrix

<table>
<thead>
<tr>
<th>Attributes</th>
<th>CSp</th>
<th>NCore</th>
<th>MSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSp</td>
<td>1</td>
<td>1/5</td>
<td>1/3</td>
</tr>
<tr>
<td>NCore</td>
<td>5</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>MSize</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

For this Matrix, the pair wise comparisons are carried out for all factors to be considered, and the matrix is completed. There is no standard way to make the pair wise comparison as this would depend on the collaborative application. For example, let Number of Processors be considered more important than Memory size. In the matrix it is rated as 5 against MSize and CSp. Next, Memory size is considered to be more important than CPU Speed and is rated as 3 against NCore and CSp.

The eigenvector called as the Priority Vector for the attributes are calculated for every attribute. The elements in each row of the matrix are multiplied with each other and then the nth root is taken for computing the priority vector values. Since, the number of attributes is 3, cubic root value is calculated and is summarized Table 4.7.

Table 4.7 Final Priority Vector of the attributes

<table>
<thead>
<tr>
<th>Attributes</th>
<th>CSp</th>
<th>Ncore</th>
<th>MSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSp</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>0.26</td>
<td>0.7</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>0.64</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

The judgment can be verified by calculating a Consistency Ratio (CR) to measure the consistency of the judgments that have been relative to
large samples of random judgments. If the CR is greater than 0.1, the judgments are untrustworthy because they are too close to randomness, pairwise comparisons have to be re-evaluated and must be repeated. The computed CR values are 0.117, 0.03 and 0.117.

Additionally, the priority vector for considering the interdependency among the attributes is calculated to be (0.11, 0.32, and 0.57). The next step is to normalize the relative values by dividing the values with their sums which is termed as Final Priority Vector. The normalized values for the judgment matrix are computed and the final priority vector for the considered sample is (0.349, 0.547 and 0.104). The value 0.547 shows that the attribute Number of Processors is given more importance; 0.349 shows that the attributes CPU speed and the Memory size is given less importance compared to Number of processors.

This system was implemented in Java and tested with the dataset generated by www.planet-lab.org from Computer Networking Research Laboratory. The system was tested for varying number of peers and for varying number of criteria. Figure 4.1 shows the results of the performance for 3, 5 and 7 criteria for 10 to 50 peers.

![Figure 4.1](image_url)  
**Figure 4.1** Performance analysis AHP method of the system for 10-50 peers
It can be seen that time taken for processing is less when the number of peers are less than 50. The difference because of the increase in number of criteria is comparatively less. However, as the number of peers increase, the time taken for processing increases proportionally to the number of criteria as shown in Figure 4.2.

![AHP method](image)

**Figure 4.2** Performance analysis AHP method of the system for 50 - 640 peers

It can be seen from the graph that initially, the time taken for decision making is more or less same for 3, 5 and 7 criteria. However, when the number of peers increases to more than 200, the time difference becomes significant and the processing time is more when the number of criteria increases. The processing time is more for AHP model because of the pairwise comparison of peers. As the number of criteria and number of peers both increase, the number of comparisons that has to be performed increases sharply.

To briefly summaries, in AHP MCDM model, the decision making was performed based on criteria, which has been defined to meet the user/application requirements. The criteria for peer selection are based on
multiple attributes like, processors, CPU speed and Memory size. The results have shown that AHP helps to arrive at an optimal decision regarding peer selection by mapping the requirements to the available resources.

4.2 PREFERENCE RANKING ORGANIZATION METHOD FOR ENRICHMENT EVALUATIONS (PROMETHEE)

PROMETHEE is a well-established decision support system which deals with the appraisal and selection of a set of options on the basis of several criteria, with the objective of identifying the pros and the cons of the alternatives and obtaining a ranking among them (Brans & Vincke 1985). PROMETHEE does not use linguistic values as AHP. Preference function based outranking method is a special type of Multi-Criteria Decision-Making (MCDM) tool that can provide a ranking ordering of the decision options. The authors have outlined two methods PROMETHEE I and PROMETHEE II (Julia et al, 2010) and (Majid et al, 2010). PROMETHEE I provide a partial pre-order while PROMETHEE II provides a total preorder on the set of possible actions. It offers the decision maker a set of actions; some of which are compared while others are not. PROMETHEE II (refereed to as simply PROMETHEE in this thesis) has been applied for the selection of peers as it possible to obtain a full ranking of the peers.

The first step PROMETHEE MCDM model is the generation of the evaluation matrix, which presents the performance of each alternative in relation to each criterion. Using the data contained in the evaluation matrix, the alternatives are compared pair-wise with respect to every single criterion. The results are expressed by the preference functions, which are calculated for each pair of options and can range from 0 to 1. The value ‘0’ means that there is no difference between the pair of options, value ‘1’ indicates a big difference.
By multiplying the preferences by the criteria weights and adding the single values, a matrix of global preferences is calculated. In this matrix, the sum of the row expresses the strength of an alternative (dominance). The sum of the column expresses how much an alternative is dominated by the other ones (sub-dominance). A linear ranking is obtained by finding the difference between the sub-dominance and the dominance value.

The PROMETHEE model is explained by considering the alternates as \( A_1, A_2, \ldots, A_n \) and the Criteria as \( C_1, C_2 \ldots C_m \). Let \( C_1, C_2 \ldots C_m \) are criteria that are used to describe the alternatives after the assignment defined as \( X_{ij} \) for the degree of alternative \( A_i \) with respect to criteria \( C_j \). Let \( W_1, W_2 \ldots W_m \) be the weight for importance of the criteria \( C_1, C_2 \ldots C_n \). The computation flow process of PROMETHEE method is stated in the following paragraphs.

Factors of importance (weight) given to each criterion. The weight values of all the criteria are defined using the Equation 4.1.

\[
W = \sum_{j=1}^{n} W_j 
\]  

(4.1)

The process of PROMETHEE model is initiated with the normalized decision matrix, which is generated using any one of the formulae given in the Equations (4.2) and (4.3) (Vijay and Shankar, 2010).

\[
R_{ij} = \frac{[X_{ij} - \text{Min}(X_{ij})]}{[\text{Max}(X_{ij}) - \text{Min}(X_{ij})]}_{(i=1,2,\ldots n, j=1,2,\ldots m)}
\]  

(4.2)

\[
R_{ij} = \frac{[\text{Max}(X_{ij}) - X_{ij}]}{[\text{Max}(X_{ij}) - \text{Min}(X_{ij})]}_{(i=1,2,\ldots n, j=1,2,\ldots m)}
\]  

(4.3)

where Max and Min specifies the maximum and minimum values of each criteria for the considered alternatives. Equation (4.3) has been used to obtain the Max and Min values for each criteria.
The difference in criteria values for each pair-wise alternative is computed and the evaluation matrix is generated using the preference function given in Equation (4.4).

\[
P_j(i, i') = \begin{cases} 
0 & \text{if } R_{ij} \leq R_{i'j} \\
(R_{ij} - R_{i'j}) & \text{if } R_{ij} > R_{i'j}
\end{cases}
\]  \quad (4.4)

The aggregated preferences function values for all the paired alternatives is computed using the Equation (4.5).

\[
\pi(i, i') = \frac{[\sum_{j=1}^{m} W_j \ast p_j(i, i')]}{\sum_{j=1}^{m} W_j}
\]  \quad (4.5)

The leaving flow expresses how much an alternative dominates the other alternatives, while the entering flow denotes how much an alternative is dominated by the other alternatives. The leaving and entering flows (multi-criteria preference flows) for different alternatives are now computed using the Equations (4.6) and (4.7) respectively. The outranking flow is the difference between the leaving and entering flows which is computed using the Equation (4.7).

\[
\varphi^+ = \frac{1}{(n-1)} \sum_{i=1}^{n} \pi(i, i') \quad (i \neq i')
\]  \quad (4.6)

\[
\varphi^- = \frac{1}{(n-1)} \sum_{i=1}^{n} \pi(i', i) \quad (i \neq i')
\]  \quad (4.7)

\[
\varphi(i) = \varphi^+(i) - \varphi^-(i)
\]  \quad (4.8)

Application of PROMETHEE MCDM technique for Peer selection is explained using an example given in Table 4.8.
The relative importance of ‘$w_j$’ and its criteria set are $C_1$ denotes the CPU Speed, $C_2$ represents Number of Cores and $C_3$ is considered to be the Free Memory space. The weightage values assumed for the criteria are $w_1 = 0.2$, $w_2 = 0.3$, $w_3 = 0.5$.

Maximum and minimum value from the sample dataset is given in the Table 4.9 and the preference functions for all pairs of alternatives for the dataset is given below.

Table 4.9 Maximum & Minimum values for the Sample Peer Data

<table>
<thead>
<tr>
<th>Max</th>
<th>3.5</th>
<th>4</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>1.5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The normalized decision matrix which is generated using the formula in the Equation (4.2) for the considered sample dataset is given in Table 4.10.

Table 4.10 Normalized decision matrix for the Sample Peer Data

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0.0833</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0.333</td>
<td>0</td>
</tr>
</tbody>
</table>
The preference functions for all pairs of alternatives are calculated using Equation 4.4.

Table 4.11 Preference Functions for the Sample Peer Data

<table>
<thead>
<tr>
<th>Location Pair</th>
<th>C₁</th>
<th>C₂</th>
<th>C₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>(1,3)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0083336</td>
</tr>
<tr>
<td>(2,1)</td>
<td>1.0</td>
<td>1.0</td>
<td>0.9166667</td>
</tr>
<tr>
<td>(2,3)</td>
<td>1.0</td>
<td>0.6667</td>
<td>1.0</td>
</tr>
<tr>
<td>(3,1)</td>
<td>0.0</td>
<td>0.3334</td>
<td>0.0</td>
</tr>
<tr>
<td>(3,2)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

The aggregated preference values for the above dataset is calculated using Equation 4.4 and is given in Table (4.12).

Table 4.12 Aggregated Preference Function Values for the Sample Peer Data

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>0.0</td>
<td>0.0833</td>
</tr>
<tr>
<td>2</td>
<td>3.25</td>
<td>-</td>
<td>2.7778</td>
</tr>
<tr>
<td>3</td>
<td>0.5556</td>
<td>0.0</td>
<td>-</td>
</tr>
</tbody>
</table>

The PROMETHEE model can give the complete pre-order by using a net flow, though it loses much information of preference relations. The entering and leaving flows for the considered dataset is computed using Equations 4.6 and 4.7 and its outranking are given in Table (4.12).
Table 4.13  Entering, Leaving and Out-Ranking Flows for the Sample Peer Data

<table>
<thead>
<tr>
<th>Rank</th>
<th>Alternate</th>
<th>Entering</th>
<th>Leaving</th>
<th>Out-Ranking Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.0278</td>
<td>0.0</td>
<td>6.0278</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.5556</td>
<td>2.8611</td>
<td>-2.3056</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.0833</td>
<td>3.8056</td>
<td>-3.7222</td>
</tr>
</tbody>
</table>

Finally, the ranking of all the considered alternatives is determined depending on the values of net-outranking flow. For the considered example, the alternative with higher value of outranking flow which is 6.0278 is chosen to be the best alternative.

PROMETHEE method was implemented in Java and tested with the same dataset as AHP. The system was tested for varying number of peers and for varying number of criteria. The results of the performance are shown in Figure 4.3 and Figure 4.4.

![Figure 4.3 Performance analysis PROMETHEE method of the system for 10 – 50 peers](Image)
Figure 4.4  Performance analysis PROMETHEE method of the system for 50 - 640 peers

Here again, as with AHP method, the time taken for processing is less when the number of peers are less than 50. The overall time taken for processing is also considerably less than the AHP method. However, when the number of peers increases to more than 50, the time difference becomes significant and the processing time is more when the number of criteria increases. If more peers are participating in the collaboration work then processing time for selection and ranking increases.

To summarize, PROMETHEE supports decision making to choose and rank peers on the basis on their resources. Unlike AHP, this method does not require any pre-processing.

4.3  ELIMINATION AND CHOICE TRANSLATING REALITY (ELECTRE)

ELECTRE is also one of the Multi Criteria Decision making (MCDM) models. This model allows decision makers to select the best choice with maximum advantage and minimum conflict in the function to multi-
attribute criteria. Different versions of ELECTRE have been developed including ELECTRE I, II, III, IV and TRI (Roy et al 1968 and Roy et al 1978). All methods are based on the same fundamental concepts but differ both operationally and according to the type of the decision problem.

The main idea of ELECTRE I is the proper utilization of “outranking relations”. Outranking relation is formed based on the dominance of relations among alternatives in this method. This method creates the possibility to model a decision making process by using two types of coordination indices. These indices are concordance and discordance matrices. The decision maker uses concordance and discordance indices to analyze outranking relations among different alternatives and to choose the best alternative using the crisp data.

The job execution application is considered to be the sample application to illustrate ELECTRE I method (Jihong et al 2011). It is assumed that $A_1, A_2 ... A_n$ are ‘n’ possible alternatives for optimal resource sharing in P2P collaboration. Let $C_1, C_2 ... C_m$ are criteria that are used to describe the alternatives after the assignment defined as $X_{ij}$ for the degree of alternative $A_i$ with respect to criteria $C_j$. Let $W_1, W_2 ... W_m$ be the weight for importance of the criteria $C_1, C_2 ... C_n$. The computation flow process of ELECTRE I method is stated in the following paragraphs. Factors of importance (weight) given to each criterion expresses the relative importance. The weight values of all the criteria are defined using the Equation 4.9 (Surendra et al 2007).

$$W = \sum_{j=1}^{n} w_j = 1$$

(4.9)

An alternative is said to dominate the other alternatives if one or more criteria exceeded (compared with the others criterion) and is equal to the
remaining criteria. Pairwise comparison of each alternative based on the criteria is denoted by ‘$x_{ij}$’. Normalization of values is applying Equation 4.10 conventionally used in matrix operations (Jihong et al 2011).

$$R = \begin{bmatrix} r_{ij} \end{bmatrix} = \frac{x_{ij}}{\sqrt{\sum_{i=m}^{i=m} x_{ij}^2}}$$ (4.10)

This weight multiplied by the matrix pairwise comparison matrix form $V_{ij}$ using the equation 4.11.

$$V_{ij} = R_{ij} * W_j = \begin{pmatrix} x_{11}w_1 & \cdots & x_{1n}w_n \\ \vdots & \ddots & \vdots \\ x_{m1}w_1 & \cdots & x_{mn}w_n \end{pmatrix}$$ (4.11)

Formation of discordance, concordance index and the index for each pair of alternatives is done through assessment of the relation ranking. For each pair of alternatives $A_f$ and $A_g$ ($f, g = 1, 2, \ldots, m$), the decision matrix for criterion $j$, the set is divided into two parts.

The set of concordance index $\{c_{fg}\}$ shows the sum of weights of criteria for which $A_f$ alternative is better than the $A_g$ alternative.

$$C_{fg} = \{ j | v_{ij} > v_{gj} \}$$ (4.12)

Where $j = 1, 2, \ldots, n$

The set of discordance index $\{D_{fg}\}$ Equation (4.13)

$$D_{fg} = \{ j | v_{ij} < v_{gj} \}$$ (4.13)

where $j = 1, 2, \ldots, n$.

Matrix of concordance ($C$) contains elements that in calculating the concordance index, and is associated with attribute weights are given in the Equation 4.14.
\[ c(f, g) = \sum_{j \neq g} W_j \]  

(4.14)

Matrix of discordance (D) contains elements of the discordance index is calculated according to (Triantaphyllou, 2000). This matrix associated with the values of attributes, namely is given in Equation 4.15.

\[ d(f, g) = \frac{\max_{j \in D_{fg}} |V_{fj} - V_{gj}|}{\max_{j, f, g \in I} |V_{fj} - V_{gj}|} \]

(4.15)

The average dominance indices are computed using the Equations 4.16 and 4.17.

\[ \overline{C}_{fg} = \frac{\sum_{f=1}^{m} \sum_{g=1}^{m} c(f, g)}{m(m-1)} \]

(4.16)

\[ \overline{D}_{fg} = \frac{\sum_{f=1}^{m} \sum_{g=1}^{m} d(f, g)}{m(m-1)} \]

(4.17)

The net superior and inferior value \( C_f \) and \( D_f \) respectively are calculated using the Equations 4.18 and 4.19. \( C_f \) sums together the number of competitive superiority for all alternatives, and the alternative with higher value is the better one. On the contrary, \( D_f \) is used to determine the number of inferiority ranking the alternatives. The smaller net inferior value gets better dominant.

\[ C_f = \sum_{i=1}^{m} c(f, i) - \sum_{i=1}^{m} c(i, g) \]

(4.18)
\[ D_j = \sum_{i=1}^{m} d(f,i) - \sum_{i=1}^{m} d(i,g) \]  

(4.19)

Application of ELECTRE technique for peer selection is explained using the sample dataset given in Table 4.7. The criteria set \( w_1 \) denotes the CPU Speed, \( w_2 \) represents Number of Cores and \( w_3 \) is considered to be the Free Memory space. The assumed weight values are 0.2, 0.5, and 0.3 whose summation is 1.

\[
\begin{pmatrix}
1.5 & 1 & 3 \\
3.5 & 4 & 25 \\
1.5 & 2 & 1
\end{pmatrix}
\]

As the initial step, the normalized matrix \( R = [r_{ij}] \) is generated using the Equation 4.9. The generated normalized matrix for the considered dataset is given as

\[
R = [r_{ij}] = \begin{pmatrix}
0.0896 & 0.0476 & 0.0047 \\
0.209 & 0.1905 & 0.0394 \\
0.0896 & 0.0952 & 0.0016
\end{pmatrix}
\]

The weighted matrix \( Vij \) is computed using Equation 4.11 and the generated weighted matrix obtained is:

\[
V = R \times W = [r_{ij}] \times w_j = \begin{pmatrix}
0.0179 & 0.0238 & 0.0014 \\
0.0418 & 0.0952 & 0.00118 \\
0.0179 & 0.0476 & 0.0005
\end{pmatrix}
\]

The coordination index Concordance (\( C_{fg} \)) and its Interval Matrix values are computed using the Equation 4.12 and the Concordance Interval Matrix is computed using the Equation 4.19.

Concordance Interval Matrix values
The Concordance Interval Matrix for the considered sample data is given as:

\[
C = \begin{pmatrix}
- & c(1,2) & \ldots & c(1,m) \\
 c(2,1) & - & \ldots & c(2,m) \\
 & \vdots & \ddots & \vdots \\
 c(m,1) & c(m,2) & \ldots & - \\
\end{pmatrix}
\]  \quad (4.19)

The coordination index Discordance \(D_{fg}\) and its interval matrix values are computed using the Equation 4.13 and the discordance interval matrix is computed using the Equation 4.20.

Discordance Interval Matrix

\[
D = \begin{pmatrix}
- & d(1,2) & \ldots & d(1,m) \\
 d(2,1) & - & \ldots & d(2,m) \\
 & \vdots & \ddots & \vdots \\
 d(m,1) & d(m,2) & \ldots & - \\
\end{pmatrix}
\]  \quad (4.20)

The discordance interval matrix for the considered sample dataset is given as:

\[
D = \begin{pmatrix}
1 & -0.1455 & 0.0397 \\
1 & - & 1 \\
1 & -0.2381 & - \\
\end{pmatrix}
\]

The average dominance indices are computed using the Equations 4.16 and 4.17 and the values \(\overline{C_{ki}}\) and \(\overline{D_{ki}}\) for the sample dataset is computed to be 0.5833 and 0.5044 respectively.
Table 4.14 shows both Net-Superior and Net-Inferior values. This value gives the ranking.

### Table 4.14 Net-Superior & Net-Inferior Values

<table>
<thead>
<tr>
<th>Rank</th>
<th>Net-Superior</th>
<th>Net-Inferior</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alternate</td>
<td>Value</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-1.2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>-0.8</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

The system was tested for varying number of peers and for varying number of criteria. The results of the performance are shown in Figure 4.5 and Figure 4.6.

![Figure 4.5](image)

**Figure 4.5** Performance analysis ELECTRE method of the system for 10 - 50 peers
Here again, as with the other methods, the time taken for processing is less when the number of peers is less than 50. The overall time taken for processing is less than the AHP method but more or less same as PROMETHEE. However, when the number of peers increases to more than 50, the time difference becomes significant and the processing time is more when the number of criteria increases. However, it can be noted from the graph that when the number of peers and criteria increase, the time taken for this method is less than PROMETHEE. The calculation process is less in ELECTRE. Hence the time consumption is also less.

Like AHP and PROMETHEE methods, it can be seen that ELECTRE method can be successfully applied to select and rank peers for collaboration activity by comparing the resource availability with the resource requirement of the collaboration applications. This method was again implemented in Java and tested with the same dataset as AHP, PROMETHEE and ELECTRE.
4.4 COMPARISON OF AHP, PROMETHEE AND ELECTRE METHODS FOR PEER SELECTION

The methods AHP, PROMETHEE and ELECTRE have been compared based on the ranking methodology, weighting factor assigned for the attributes given as requirements by the peers and other criteria.

It has been seen that all the three methods can be used for selection of the right peer for collaboration. However, there are differences in the approach used by the three methods. The comparative analysis of the three methods is given in Table 4.15.

Table 4.15 Comparison between AHP, PROMETHEE and ELECTRE

<table>
<thead>
<tr>
<th>Ranking Method/Properties</th>
<th>AHP</th>
<th>PROMETHEE</th>
<th>ELECTRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking Methodology</td>
<td>Peers are compared and ranked on the basis of the required value of the attributes</td>
<td>Peers are compared and ranked based on the maximum available value of the attributes</td>
<td>Peers are compared and the ranked based on the maximum available value of the attribute</td>
</tr>
<tr>
<td>Weighting factor</td>
<td>Weighting factor is given for the whole set of attributes used for ranking</td>
<td>Weighting factor is assigned for individual attributes</td>
<td>Weighting factor is assigned for individual attributes</td>
</tr>
<tr>
<td>Preprocessing for pair wise comparison</td>
<td>Required</td>
<td>Not required</td>
<td>Not required</td>
</tr>
<tr>
<td>Pair wise comparison</td>
<td>Pair wise comparison</td>
<td>Partial pair wise comparison</td>
<td>Partial pair wise comparison</td>
</tr>
<tr>
<td>Computational Speed</td>
<td>Slow</td>
<td>Medium</td>
<td>Fast</td>
</tr>
</tbody>
</table>
AHP method requires preprocessing to convert the set of discrete linguistic choices available to the decision maker to a discrete set of numbers representing the importance or weight of linguistic choices. Hence this increases the processing time for AHP model. As the number of peers and/or number of criteria increases, the processing time increases sharply. This can also be clearly observed from the performance results shown in Figures 4.7, 4.8 and 4.9 for 3, 5 and 7 criteria respectively.

Figure 4.7 Performance analysis of 3 criteria's 50-640 Alternates

Figure 4.8 Performance analysis of 5 criteria's; 50-640 Alternates
It can be observed that while the processing time for AHP increases sharply with increase in criteria and number of peers, the time taken by PROMETHEE and ELECTRE methods is more or less the same. Overall, it was observed that when the number of peers was less, the processing time was more or less similar for all methods for varying number of criteria. However, when the number of peers increases, the time taken for processing increases, but is more for AHP and significantly less for PROMETHEE and ELECTRE. Also AHP method requires preprocessing steps as compared to that of the other methods which also leads to increase in processing time.

AHP uses matrix algebra to sort out factors to arrive at a mathematically optimal solution (Melvin 2012). Hence for peer selection the weighting factor is applied for the whole set of resource attributes. In PROMETHEE and ELECTRE, however weighting factor of attributes is individually applied. Hence in PROMETHEE and ELECTRE the decision making process selects peers based on the highest value of the attribute while AHP method goes for the optimal value with respect to the application.
All of the above methods are tried and tested methods for multi-criteria decision making. The selection process is applied for the whole set of resources for all the participating peers. In a P2P network, the resource availability dynamically changes and peers also join and leave the network dynamically. Hence, it is necessary to explore methods where the time taken for decision making is considerably less.

4.5 CONCLUSION

The process of selecting the right peer for collaboration requires decision to be made on multi criteria. Traditional MCDM models AHP, ELECTRE and PROMETHEE have been applied for Peer collaboration. The models have been implemented and tested performance for different criteria and for different number of criteria has been presented. The set of alternate peers have been determined and the peers have been ranked according to their resource availability. The comparative performance of the three models has been presented and discussed.